

NASA-TM-84253 19820018413

A Floating-Point/Multiple-Precision Processor for Airborne Applications

Robert Yee

FOR REFERENCE

NOT TO BE TAKEN FROM THIS ROOM

February 1982

LIBRARY COPY

JUN 27 1982

LANGLEY RESEARCH CENTER
LIBRARY, NASA
HAMPTON, VIRGINIA



National Aeronautics and
Space Administration

ENTER:

DISPLAY 82N26289/2

82N26289*# ISSUE 17 PAGE 2356 CATEGORY 6 RPT#: NASA-TM-84253

A-8832 NAS 1.15:84253 82/02/00 11 PAGES UNCLASSIFIED DOCUMENT

UTTL: A floating-point/multiple-precision processor for airborne applications

AUTH: A/YEE, R.

CORP: National Aeronautics and Space Administration, Ames Research Center,
Moffett Field, Calif. AVAIL.MTIS SAP: HC A02/MF A01

MAJS: /*AIRBORNE/SPACEBORNE COMPUTERS/*FLOATING POINT ARITHMETIC/*INPUT/OUTPUT
ROUTINES/*MICROCOMPUTERS

MINS: / AIR NAVIGATION/ AIRCRAFT CONTROL/ AVIONICS/ DIGITAL COMPUTERS/ FLIGHT
TESTS/ MICROPROCESSORS/ VERTICAL TAKEOFF AIRCRAFT

ABA: S.L.

ABS: A compact input output (I/O) numerical processor capable of performing
floating-point, multiple precision and other arithmetic functions at
execution times which are at least 100 times faster than comparable
software emulation is described. The I/O device is a microcomputer system

bit registers running at a 5 MHz clock rate, 18K random access memory

(RAM) and 4K electrically programmable read-only memory (EPROM). The

2 M

A Floating-Point/Multiple-Precision Processor for Airborne Applications

Robert Yee, Ames Research Center, Moffett Field, California



National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

[#]
N82-26289

A FLOATING-POINT/MULTIPLE-PRECISION PROCESSOR
FOR AIRBORNE APPLICATIONS

Robert Yee
Ames Research Center, NASA
Moffett Field, CA 94035

ABSTRACT

The development of Vertical Take-off or Landing (VTOL) research programs in the areas of guidance, control, navigation and instrumentation aboard a research helicopter demonstrated a limitation characteristic of some digital flight-control computers: a lack of hardware floating-point processing. This limitation restricts the implementation of wide dynamic-range variables and recursive digital-filtering functions where high precision and speed are required.

This paper describes a compact Input/Output (I/O) numerical processor capable of performing floating-point, multiple-precision and other arithmetic functions at execution times which are at least 100 times faster than comparable software emulation. The I/O device is actually a microcomputer system containing a 16-bit microprocessor, a numerical coprocessor with eight 80-bit registers running at a 5 MHz clock rate, 18K Random Access Memory (RAM) and 16K Electrically Programmable Read Only Memory (EPROM). The processor acts as an intelligent slave to the host computer and can be programmed in high-order languages such as FORTRAN and PL/M-86.

The I/O interface between the numerical processor and the host computer is a pseudo-Direct Memory Access (DMA) that allows asynchronous operations during parallel data and instruction transfers. The I/O interface techniques described herein can be incorporated to accommodate host computers other than those used by the author.

INTRODUCTION

A Bell UH-1H helicopter, equipped with a fly-by-wire flight control system and down-link telemetry, serves as a flying test bed for research activities involving advanced avionics, air data sensors, navigation, guidance and control for VTOL aircraft. The research activities are coordinated aboard the helicopter via a pair of digital flight computers. As flight tests aboard the UH-1H developed in complexity, the demands on the flight computers became more sophisticated.

An evaluation was conducted to explore the possibility of enhancing the existing computers with floating-point and multiple-precision capabilities. Because of the software and hardware complexities, modification of the existing computers by register extension was ruled out. It became obvious that an I/O peripheral device capable of performing floating-point and multiple-precision numerical operations was needed. To meet the requirements of physical space, temperature extremes and programming flexibility, the processor had to be a relatively compact unit limited to military-standard components, be programmable, and allow asynchronous I/O transfers.

Although some of the requirements were severe and at times conflicting, this paper will demonstrate how they are being satisfied with a minimum amount of space and fabrication difficulties.

THE REQUIREMENTS

During the course of VTOL research and development on a simulator, desired outputs for a unit step input on flight-control algorithms with long time constants produced an output resembling a second-order overdamped response, instead of the expected critical damped response (Figure 1). Investigation into the problem showed that the undesired response was the result of truncation and rounding errors by the flight computers which did not have sufficient fixed point resolution to handle the accuracy required.

This problem of insufficient fixed point resolution also appeared when digital filters with recursive functions were attempted on the simulator. Since the flight computers lacked floating-point capabilities, the possibility of employing double words was explored. Unfortunately, software complexities and slow execution times prohibited this option.

It became apparent that an I/O device capable of performing floating-point (for wide dynamic range) or multiple-precision (for high accuracy) sub-routines was required. The I/O device had to be a self-contained programmable numerical processor to avoid tying up the host computer with excessive

I/O instruction transfers. Since the numerical processor would be required to fly aboard the UH-1H helicopter, all integrated circuit components had to satisfy MIL-STD-883 Level B requirements.

THE PROCESSOR

After an examination of the requirements, a (SECS 86/05 microcomputer system with 16K RAM, 16K EPROM and the Intel 8087 numeric coprocessor was selected. The microcomputer system is based upon the 8086 microprocessor and is *Multibus compatible. An I/O interface between the *Multibus and an existing flight computer I/O port is being designed and fabricated at Ames Research Center, the details of which will be discussed later in this paper.

The microcomputer system which the author refers to as the floating-point/multiple-precision processor (or "processor" for abbreviation) is a militarized version of the Intel iAPX86 Central Processing Unit (CPU) based system. The entire system fits into a 6-slot motherboard enclosed in a short 1/2 Air Transport Radio (ATR) chassis. Two slots are occupied by a 115 VAC @ 47-440 Hz power supply and two additional slots are occupied by the CPU and memory boards. The fifth slot is taken up by the I/O interface and the sixth slot is a spare. The system, with the exception of the I/O interface board, is available through the Severe Environment Systems Company (SESCO).

At the heart of the microcomputer system is the Intel 8087. The 8087 Numeric Data Processor (NDP) is a 40-pin Metal Oxide Semiconductor (MOS) device that is wired parallel to the 8086 with both operating at a 5 MHz clock rate. The NDP serves as a numeric coprocessor to the 8086 by extending the 8086 registers with the addition of over 50 instructions to the 8086 instruction set. The NDP and the 8086 fetch and decode instructions in parallel, arbitration between the two is handled automatically via their respective handshaking lines (Figure 2). Thus the programmer need not treat the 8087 NDP as an I/O device to the 8086; the NDP is transparent to the programmer.

The 8087 expands the 8086 data type to include 32-, 64-bit integers, 32-, 64-, 80-bit floating-point and 18-digit Binary Coded Decimal (BCD) operands. A more detailed explanation of the data formats and types is shown in Figure 3 and Table 1. The NDP directly extends the 8086 instruction set to include trigonometric, exponential, logarithmic, square root, addition, subtraction, multiplication and division operations for all data types. Additional features of the NDP include rounding control and maskable exception (i.e., zero divide) handling. Exception handling, if unmasked, produces a programmable interrupt to the 8086.

All data types are automatically converted to an 80-bit floating point format (temporary real data type). This format is used by the 8087 for all internal operations, thus shielding the final results from the effects of rounding and overflow/underflow in intermediate calculations.

*All mnemonics copyright Intel Corporation 1980.

Representative execution times are given in Table II. 8087 execution times are at least 100 times faster than 8086 software emulation, and about 10 times faster than other comparable MOS devices. Although bipolar multipliers are faster than the NDP, these devices are extremely limited in hardware arithmetic operations and data types.

I/O INTERFACE

In the configuration presented so far, the NDP readily communicates with the *Multibus. Unfortunately, the existing flight computers are not *Multibus compatible. Two paths were evaluated to remedy this problem: the "front-door" and "back-door" approaches.

The "front-door" approach was to design an interface which would allow the flight computers to directly communicate with the *Multibus. This approach showed timing, hardware and software problems. The two computer systems operate at different clock speeds (the flight computer @ 10 MHz and the 8086 @ 5 MHz), and the number of integrated circuits required might exceed the space available on the I/O board. The two computer systems have vastly different instruction sets, therefore developing a cross-assembler between the two computers would not be cost effective.

In light of the problems encountered with the front-door approach, a discrete entry into the back-door via an external RAM was selected. The problems were reduced into their simplest components. If the two computer systems operate at different clock rates, why not have them operate asynchronously? This configuration allows the 8086-based system to be preprogrammed in its own language with numerical subroutines. The flight computer's I/O port would transfer data, select the appropriate subroutines, and receive the processed data. The first word sent to the 8086-based system is an address word that selects the appropriate subroutine from the *Multibus EPROM memory, followed by the data words loaded into RAM. The 8086-based system activates the NDP as needed, performing arithmetic and logic functions with its own instruction set, and transferring the results back to RAM. This eliminates the need for a cross-assembler. The problem of interfacing between the two systems is reduced by both systems sharing the same RAM (Figure 4).

The *Multibus readily accommodates external RAM locations, but this is not true of the flight computer's I/O port which lacks address generation. This deficiency was overcome by a pair of Advanced Micro Devices (AMD) 2940 address generators. The I/O interface, as shown in Figure 4, allows the flight computer I/O command and data lines to gain access to the *Multibus via the external RAM (located on the I/O interface board).

The I/O interface contains an external memory which behaves as RAM locations as far as the *Multibus is concerned, but is regarded as a stack for the flight computer's I/O port. Every time a flight computer transfers or receives data from the external memory, the address generators increment or decrement their internal base address and word

count registers. The base address and word count are selected by the 8086-based system. The external RAM's write enable, output enable, and chip select lines are logically OR'ed with the appropriate command lines from the *Multibus and the flight computer's I/O control lines. Data and address lines from the *Multibus, the address generators, and the flight computer are isolated by tri-state buffers.

Since the data and address lines of both systems are tri-stated from each other, both are free to operate asynchronously as long as they do not try to access the external RAM at the same time. To avoid bus conflicts, the I/O interface takes advantage of the address generator ability to output an active high signal (DONE) when the word counter reaches zero (completion of data and instruction transfers). The utilization of the DONE signal for bus arbitration is illustrated in the following I/O transfer sequence:

1. The 8086 loads a base address to the address generators and sets the word counter to the number of data words being transferred from the flight computer. The DONE signal is now low.
2. The 8086 sets a latch on the I/O request line of the flight computer for data transfer into the external RAM at the address selected by step 1.
3. If the data request is not masked by the flight computer software, the flight computer will output data words into the external RAM. Each data word transfer will increment the base address of the address generator and decrement the word the word counter until a zero word count is established.
4. While waiting for completion of data transfer into the external RAM from the I/O port of the flight computer, the 8086 is free to perform other programs until an interrupt occurs.
5. The word counter, being zero, drives the DONE signal high when the flight computer completes data transfer. The high signal sets a latch to interrupt the 8086 and clears the data request line to the flight computer. The flight computer is now free to execute other tasks.
6. If unmasked, the 8086 acknowledges the interrupt, clears the DONE signal and enters into a subroutine from the EPROM to evoke the NDP. The subroutine selection could be determined by a previous I/O instruction transfer from the flight computer or be contained in one of the data words transferred.
7. Upon completion of the subroutine, the results are stored in the external RAM.
8. The 8086 loads a new base address to the address generator and sets the word counter to the number of resultant data words stored in RAM. The DONE signal is still low at this step.

9. The 8086 sets a latch on the I/O request control line of the flight computer for data transfer from the external RAM at the address selected by step 8.

10. The 8086 is now again free to perform other assigned tasks.

11. The flight computer inputs the results to its data buffer. Each data transfer increments the address generator and decrements the word counter. When the word count is zero, the DONE signal is high and the request is cleared.

12. Steps 1 to 11 are repeated each time the flight computer desires the 8086-based system to perform a numeric subroutine.

Rather than interrupt the 8086-based system upon completion of data transfer into the external memory, the DONE signal can be latched to the input of a tri-state buffer. The output of the buffer is tied to a *Multibus data line. This allows the 8086 to read the status of the DONE signal. This option allows the processor to complete an assigned numeric subroutine, read the status of the DONE signal, and enter into a status loop waiting for the next subroutine request, thus avoiding time-consuming interrupts. Both the interrupt and read status options are software selected.

All data stored in the external RAM are in the form of 16-bit words and stored at even address boundaries. The lowest address bit (A0) on the *Multibus is used by the decoding circuitry to "awake" the I/O interface board when A0 is low (even address). The next 11 address bits (*Multibus A1 to A11) are logically OR'ed with the lines of the address generator to the RAM. All locations in RAM are even addresses containing 16 bits of data. This scheme takes advantage of the ability of the *Multibus to access RAM in one memory cycle versus two cycles if data were contained at odd address boundaries.

SUMMARY AND CONCLUSIONS

Flight computers without floating-point or multiple-precision hardware are severely limited in applications requiring recursive functions or wide-dynamic variables. Software overhead and execution times to handle double words, overflow/underflow, truncation and rounding errors during intermediate calculations usually restrict these applications to situations where the combination of high speed and accuracy are not required. At Ames Research Center this problem is being approached by developing a compact numeric processor, external to the flight computer. The flight computer regards the processor as an intelligent I/O peripheral capable of performing subroutines requiring floating-point or multiple-precision arithmetic.

The floating-point/multiple-precision processor is a *Multibus system with an 8086 CPU and 8087 coprocessor. The 8087 effectively extends the instruction set of the 8086 and registers for numeric operations on various data types at execution times between 17 to 100 microseconds. Programs can be written in higher-order languages such as FORTRAN OR PL/M-86.

*All mnemonics copyright Intel Corporation 1980.

Data transfer between the processor and the flight computer is accomplished by a pseudo-DMA technique, a variation of a dual port RAM. This technique simplifies integrating the two unique bus systems by reducing the integration to both systems handshaking the same external memory. This arrangement allows the 8086-based system to treat the external memory as an extra 2K X 16 RAM on the *Multibus while the flight computer regards the same memory as I/O stacks. The external memory serves as an input and output port to both systems.

The first word transferred by the flight computer is a *Multibus EPROM address that performs a desired numeric subroutine on data that are subsequently stacked on the external memory. Upon completion of storing the resultants into the external memory, the 8086-based system initiates the I/O transfer request lines to the flight computer. The I/O technique allows asynchronous operation between the two systems by incorporating a programmable word counter which outputs a signal to denote completion of data transfer into the external memory. The word counter frees both systems from having to constantly monitor the status of I/O transfer.

Since the flight computer can load data directly into the *Multibus memory, bypassing the accumulator of the 8086, the minimum I/O transfer time is limited by the access times of the external memory (read and write) or the handshaking execution times of the flight computer. The 8086 or 8087 can perform a read or write on the external memory in

approximately 2 microseconds while the flight computer performs the same task in 1.7 microseconds.

The floating-point/multiple-precision processor with the I/O interface circuitry is contained within a short 1/2-ATR chassis. The processor can be fabricated to meet the military-standard environment. The only exception is the 8087 coprocessor; a militarized version is scheduled to be available by the end of 1982.

ACKNOWLEDGMENT

The author wishes to extend his sincere gratitude for the guidance and support from the following people: David Brocker, Gilbert Robinson, John Maher, Steve Kaczmarek, Nels Watz, Mahlon Lee, John Stickley, James Ahlman, Gayle Kirby, Willis McNeil, and Andrew Kirk.

REFERENCES

Intel Corporation, "The 8086 Family User's Manual - Numerics Supplement," July, 1980.

Intel Corporation, "iSBC 86/12 Single Board Computer Hardware Reference Manual," 1978.

Covert, Guy D., and Eldon, John, "Floating-Point Adder Chip Fills Digital-Processing Gap," Electronic Design, Vol. 29, No. 24, Nov., 1981, pp. 187-192.

*All mnemonics copyright Intel Corporation 1980.

TABLE I
Data Types

Data Type	Bits	Significant Digits (Decimal)	Approximate Range (Decimal)
Word integer	16	4	$-32,768 \leq X \leq + 32,767$
Short integer	32	9	$-2 \times 10^9 \leq X \leq + 2 \times 10^9$
Long integer	64	18	$-9 \times 10^{18} \leq X \leq + 9 \times 10^{18}$
Packed decimal	80	18	$-99 \dots 99 \leq X \leq + 99 \dots 99$ (18 digits)
Short real*	32	6-7	$8.43 \times 10^{-37} \leq X \leq 3.37 \times 10^{38}$
Long real*	64	15-16	$4.19 \times 10^{-307} \leq X \leq 1.67 \times 10^{308}$
Temporary real	80	19	$3.4 \times 10^{-4932} \leq X \leq 1.2 \times 10^{4932}$

*The short and long real data types correspond to the single and double precision data types defined in other Intel numerics products.

Reprinted by permission of Intel Corporation, Copyright 1980.

TABLE II
8087 Emulator Speed Comparison

Instruction	Approximate Execution Time (μ s) (5 MHz Clock)	
	8087	8086 Emulation
Multiply (single precision)	19	1,600
Multiply (double precision)	27	2,100
Add	17	1,600
Divide (single precision)	39	3,200
Compare	9	1,300
Load (single precision)	9	1,700
Store (single precision)	18	1,200
Square root	36	19,600
Tangent	90	13,000
Exponentiation	100	17,100

Reprinted by permission of Intel Corporation, Copyright 1980.

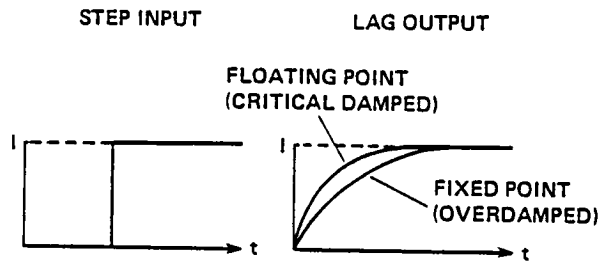


Figure 1: Lag Response Output

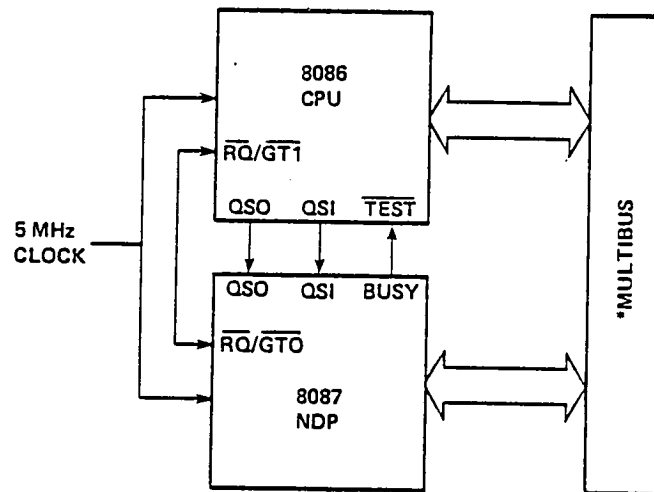


Figure 2: NDP Interconnect (Reprinted by permission of Intel Corporation, Copyright 1980)

NOTES:

- S = SIGN BIT (0 = POSITIVE, 1 = NEGATIVE)
- d_n = DECIMAL DIGIT (TWO PER BYTE)
- X = BITS HAVE NO SIGNIFICANCE; 8087 IGNORES WHEN LOADING, ZEROS WHEN STORING.
- Δ = POSITION OF IMPLICIT BINARY POINT
- I = INTEGER BIT OF SIGNIFICAND; STORED IN TEMPORARY REAL, IMPLICIT IN SHORT AND LONG REAL

EXPONENT BIAS (NORMALIZED VALUES):

SHORT REAL: 127 (7FH)
LONG REAL: 1023 (3FFH)
TEMPORARY REAL: 16383 (3FFFH)

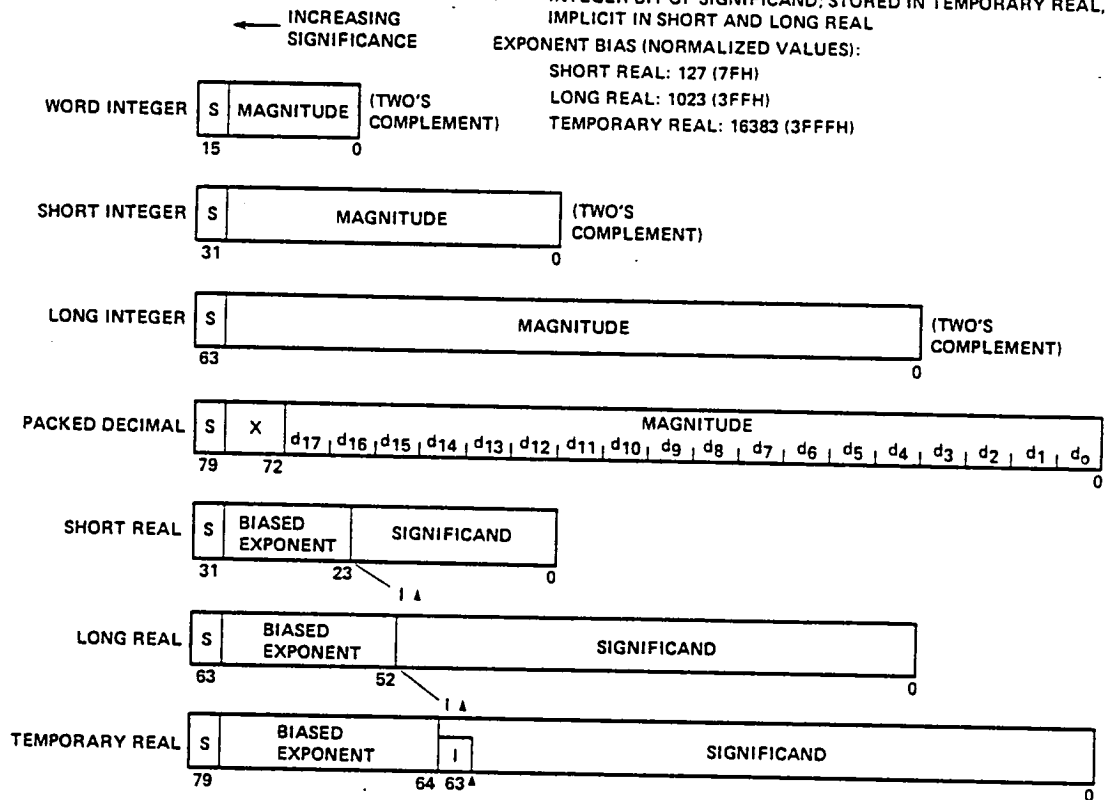


Figure 3: Data Formats

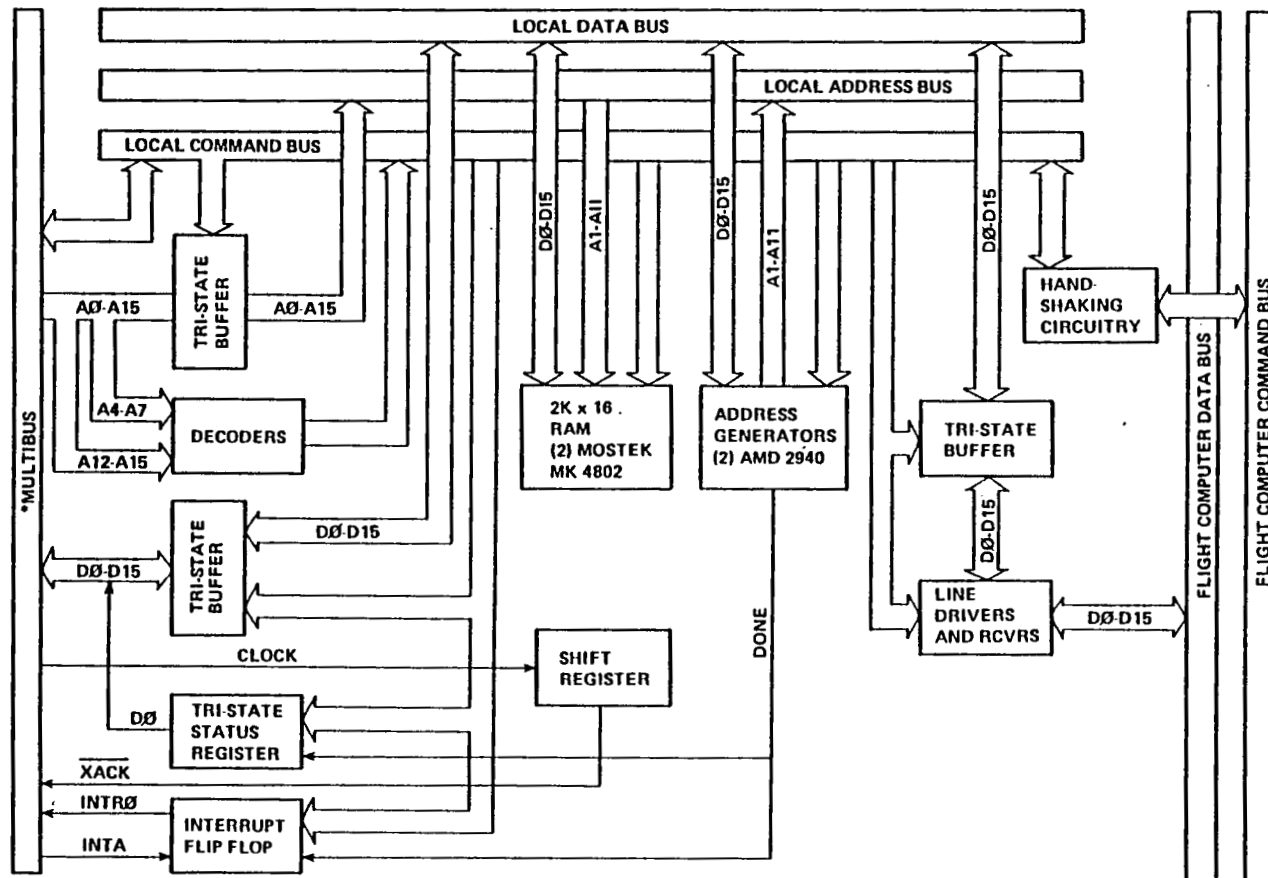


Figure 4: I/O Interface Architecture

1. Report No. NASA TM 84253		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle A Floating-Point/Multiple-Precision Processor for Airborne Applications				5. Report Date February 1982	
				6. Performing Organization Code	
7. Author(s) Robert Yee				8. Performing Organization Report No. A-8832	
9. Performing Organization Name and Address NASA Ames Research Center Moffett Field, California 94035				10. Work Unit No. T-5844	
				11. Contract or Grant No.	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546				13. Type of Report and Period Covered Technical Memorandum	
				14. Sponsoring Agency Code 532-06-11	
15. Supplementary Notes Point of Contact: Robert Yee, Ames Research Center, M.S. 210-5, Moffett Field, California 94035. Phone: (415) 965-5488, FTS: 448-5488.					
16. Abstract The development of Vertical Take-off or Landing (VTOL) research pro- grams in the areas of guidance, control, navigation and instrumentation aboard a research helicopter demonstrated a limitation characteristic of some digital flight-control computers: a lack of hardware floating-point processing. This limitation restricts the implementation of wide dynamic- range variables and recursive digital-filtering functions where high precision and speed are required. This paper describes a compact Input/Output (I/O) numerical processor capable of performing floating-point, multiple-precision and other arith- metic functions at execution times which are at least 100 times faster than comparable software emulation. The I/O device is actually a microcomputer system containing a 16-bit microprocessor, a numerical coprocessor with eight 80-bit registers running at a MHz clock rate, 18K Random Access memory (RAM) and 16K Electrically Programmable Read Only Memory (EPROM). The pro- cessor acts as an intelligent slave to the host computer and can be pro- grammed in high-order languages such as FORTRAN and PL/M-86. The I/O interface between the numerical processor and the host com- puter is a pseudo-Direct Memory Access (DMA) that allows asynchronous operations during parallel data and instruction transfer. The I/O inter- face techniques described herein can be incorporated to accommodate host computers other than those used by the author.					
17. Key Words (Suggested by Author(s)) Floating Point Military Standard 16-Bit Microprocessor				18. Distribution Statement Unlimited Subject Category 06	
19. Security Classif. (of this report) Uncl		20. Security Classif. (of this page) Uncl		21. No. of Pages 10	
				22. Price* A02	



3 1176 00504 4400